Learning of Highly-Filtered Data Manifold Using Spectral Methods*

Oleg Roderick and Ilya Safro

Argonne National Laboratory, Argonne, IL, USA
(roderick,safro)@mcs.anl.gov

Abstract. We propose a scheme for improving existing tools for recovering and predicting decisions based on singular value decomposition. Our main contribution is an investigation of advantages of using a functional, rather than linear approximation of the response of an unknown, complicated model. A significant attractive feature of the method is the demonstrated ability to make predictions based on a highly filtered data set. An adaptive high-order interpolation is constructed, that estimates the relative probability of each possible decision. The method uses a flexible nonlinear basis, capable of utilizing all the available information. We demonstrate that the prediction can be based on a very small fraction of the training set. The suggested approach is relevant in the general field of manifold learning, as a tool for approximating the response of the models based on many parameters. Our experiments show that the approach is at least competitive with other latent factor prediction methods, and that the precision of prediction grows with the increase in the order of the polynomial basis.

1 Introduction

The ability of predict outcomes of various model-constrained decisions is important in many fields of data mining. Many theoretical and applied problems, such as biological data mining, organization and cataloguing of media, policy and stock planning, optimal social networking, physical models factor analysis and uncertainty quantification, customer recommendation services require an automatic procedure that learns on the available data and provides efficient prediction of the model response, or recovery of missing data.

A variety of modern approaches to such problems are based on singular value decomposition (SVD) and on the extraction of a limited number of latent factors. These methods successfully address the two important issues: scalability and quality of the prediction. In many approaches, these issues depend on each other and, thus, pose trade-off questions [1,2]. Matrix factorization (or a solution of an eigensystem) is a basic computationally expensive step associated with these methods. Therefore, reducing the amount of data (or the size of training set) is important for the development of modern data mining systems [3,4,5].

In our work, we propose and demonstrate a method that improves existing tools for recovery and prediction of decisions based on SVD and latent factors methods. The

* This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

C. Blum and R. Battiti (Eds.): LION 4, LNCS 6073, pp. 154–168, 2010.

novelty of the method is twofold: its generalization of a widely used *linear representation* of the relationship between the training set and decisions, and its construction of an adaptive *high-order polynomial interpolation scheme* to estimate the relative probability of each possible outcome of a decision. Our method uses a flexible nonlinear basis, capable of utilizing all the available information. At the same time, the prediction can be based on a very small fraction of the training set. We show how a successful prediction can be done after randomized removal of 85% of the previously known information, whereas the linear method suffers from introducing such filtering. Another attractive feature of our method is that it treats decision outcomes as *events*, rather than *real-valued outputs* of some functional. This leads to better understanding of the training set, such as correct interpretation of gappy data, and of ordered sets of outcomes. We also show that the precision of prediction grows with the increase in the order of the polynomial basis, thereby justifying the use of high-order interpolation schemes. The complexity of the proposed method is equivalent to the complexity of partial spectral decomposition of the training data. The proposed approach is beneficial for data mining and prediction problems when there are reasons to expect nonlinear relationships between model response and inputs.

1.1 Generalized Problem and Assumptions

Consider a situation in which a group of decision-makers (*users*) is given access to a set of similar media or products (*content*). The users evaluate some of the units of content by assigning one of the suggested tags, or *ratings*, to each unit. Given samples of rated content, we seek to recover, or predict, the ratings that are not available. Applied problems of this type include ranking scientific papers by predicted preferences of a researcher (Google scholar-type search, [6]), predicting whether two users of a social networking service will be interested in establishing a connection (LinkedIn-type search, [7]), and predicting ratings that customers assign to movies offered for purchase or rent (MovieLens database, Netflix problem, [8]). For additional information, we refer the reader to such sources as [9,10,11].

The proposed scheme is demonstrated on the data set taken from the Netflix problem [8]. Netflix provides a large example database and has a large community of contest participants trying to improve the performance of predictions on that example by a certain objective quantity. While the Netflix contest is not necessarily the most general problem in the field, the availability of the training set and objective comparisons of performance has prompted us to use it as our test case. To the best of our knowledge, the best strategies represent a mixes of many different approaches. We believe that no *one* particular algorithm is able to predict decisions (up to the precision desired by the contest) under a large amount of uncertainty. Thus, we concentrate on improving one specific family of methods (based on SVD and latent factors) that was used by many participants.

Our general question is how to predict the response of a model that is dependent on many inputs. The exact effects of some of the the input data on the model can be inaccessible, or too complicated to be examined directly. The response is then estimated based on the available observations of the model behavior for a set of inputs. In this context, our work is of interest to the field of manifold learning (e.g. [12,13]), where the general task is to discover the unknown relationship

$$\mathcal{F}(q) : T \cup Q \to \mathbb{R}\,, \tag{1}$$

where T and Q are the training set and the query set, respectively, and q is a particular point. In many methodologies, a mathematical response model $F \approx \mathcal{F}$ is constructed so that the known decisions are reproduced almost perfectly, and the deviation from truth for the unknown decisions is minimal in some sense. Schematically,

$$q \in T : F(q,T) = \mathcal{F}(q), \text{ almost surely, and}$$

$$q \in Q : F(q,T) - \mathcal{F}(q) \text{ is minimal.}$$

The following considerations motivated our approach. The unknown relationship $\mathcal{F}$ is traditionally described in SVD-based methods by a linear approximation. At the same time, in modeling the potentially nonlinear effects, a polynomial interpolation provides higher quality. A similar method is described below.

1.2 Related Work: Estimating Uncertainty Effects by Polynomial Regression

The approach to prediction and data recovery presented in this paper is an extension of recent advances made in estimation of the complex physical model response to uncertain inputs. In the available literature [14], the method was applied to a deterministic model of the nuclear reactor core, taking into account multiple physical effects resulting in complex nonlinear dynamics with unknown correlations between the parameters. For such problems, the range of values for the output is typically predicted either by Monte Carlo methods or by *linear approximations* (which is an aspect we improve in this paper). It turns out that a polynomial approximation is superior in quality and can be constructed computationally efficiently by using a very small sample of model states. The approximation reproduces both the distribution and the global sensitivities of the output, with an order of magnitude improvement over the performance of the linear model [15].

The prediction of response of physical systems to stochastic parameters has structural differences from the more general question considered in this paper. For example, for physical problems, the allowed training set is typically very small, the relevant prediction-influencing factors are already determined and relatively few. At the same time, the general task and the implied sequence of steps are esentially the same. Both problems require extracting the most important factors from a larger list, constructing a high-quality approximation of the response, and demonstrating its superiority in comparison with a traditional linear approximation. Thus, the success of the related work has prompted our interest in applying the method to a general area of model response prediction.

2 High-Order Interpolation Decision Function

Latent factor and SVD-based approaches to automatic prediction mostly have the same structure on the basic theoretical level. We find that many methods amount to a form of linear regression [16,17]. As a contribution to a field with a perceived advantage of widespread linear methods, we suggest a general, easily implemented, unsupervised approach that is neither limited to linear approximation nor derives its quality from weighted averaging of multiple linear approximations.

2.1 Intuition Behind the Method

We refer to one of the basic approaches to predicting ratings as a background for our suggestions. *Latent factor methods* [18] estimate the response of an unknown function based on a small number of quantities derived from the training set. Such a quantity, or a latent factor, is a linear or a more general functional combination of data. Latent factors are extracted from the training set by using essentially data compression techniques [19,20]. From this point we assume that the reader is familiar with the basic notion of latent factors; the relevant textbook information can be obtained at [18,19,20].

Suppose that the most important factors $S = (s_1, s_2, ..., s_k)$ are used to approximate the response to query q:

$$F(q, T) \sim F'(q, S),$$

where $S = S(T)$. Different applied problems and different approaches may call for various assumptions on how to extract S from T. For example, we can assume that a latent factor is primarily an additional characterization of a fixed user (i.e., a quantity of classification used to describe this user). Then this factor will be extracted based on a portion of the training set that includes all the decisions by this user, but perhaps not the decisions of all other users. Our goal is to introduce an approach to construct a class of decision-making functions θ based on subsets of latent factors. The members of this class will be "goal-oriented", in other words, for different types or families of queries it will simulate a work of different F' and S. The function θ will be constructed as an explicit polynomial expression on a set of factors, that is, an expansion in the polynomial basis with the coefficients obtained by regression on the training set.

The scheme used to test our approach is presented in Algorithm 1. Each step will be described in Section 3.

Algorithm 1. General Scheme

1. Application-oriented preprocessing
2. Randomized filtering of the training set
3. Latent factors identification by SVD
4. Estimated probabilities of decisions ←
   polynomial approximation on latent factors
5. Application-oriented postprocessing

3 Explanation of the Method

Let $M = \{m_i\}_{i=1}^{N}$, $U = \{u_i\}_{i=1}^{n}$ and $R = \{r_i\}_{i=1}^{K}$ be the sets of content, users and available ratings, respectively. Denote by $\mathcal{R}$ the binary approval of the real rating assignment (1 for "approves some r_i for the content", 0 for "does not approve some r_i content"):

$$\mathcal{R} : U \times M \times R \to \{0, 1\}. \tag{2}$$

To meet our goal, the decision-making function θ, needs to be able to estimate a likelihood of each rating:

$$\begin{aligned} &\theta : U \times M \times R \to (-\epsilon, 1+\epsilon) \text{ or} \\ &\theta(u_i, m_j, r_k) \quad = \xi_r \ , \end{aligned} \tag{3}$$

where $\xi_r \approx 1$ corresponds to a highly likely rating r_k and $\xi_r \approx 0$ corresponds to an unlikely rating r_k. A sufficiently small distortion ϵ is allowed to account for the interpolation error and other numerical effects.

We suggest two simplified options for constructioning of θ: *content-* and *user-centered* constructions. According to the content-centered construction, a decision of u_i on m_j is a function of decisions of u_i on other content. For the user-centered assumption, a decision of u_i on m_j is a function of other users decisions on m_j. These options are not mutually exclusive, and can, in principle, be combined. Our choice is based on the dimensions and density of the available training set. From this point, the experimental results, explanation of method, and, hence, the θ construction will be user-centered, since it provides a larger portion of the training set with a more uniformly distributed sparsity. Preliminary experiments with the goal-centered construction led to results of similar quality but worse running time.

The main problem of this paper is formulated as follows.

Problem 1. Given u_i, r_k, and a subset of content $M_i \subseteq M$, construct a user-centered θ such that

$$\sum_{m \in M_i} \|\theta(u_i, m, r_k) - \mathcal{R}(u_i, m, r_k)\|_2$$

is minimized.

3.1 Problem-Oriented Preprocessing

Given the conditions of Problem 1, denote by $\Omega_k \in \{0,1\}^{N_i \times n}$, where $N_i = |M_i|$, a binary matrix extracted from a training set as

$$\Omega_k(p,q) = \mathcal{R}(u_p, m_q, r_k). \tag{4}$$

As a preprocessing step, one has to prepare Ω_k per r_k that will participate in the final decision postprocessing. If the value $\mathcal{R}(u_p, m_q, r_k)$ is not available, then $\Omega_k(p,q) = 0$, with the exception of the position corresponding to the query that is currently addressed. We denote it by entry 1, to avoid the implication that every rating r_k is extremely unlikely.

The division of data with different numerical ratings into several binary fractions is explained by the relationship between the numerical values and missing entries. In frequency matrices of usual information retrieval problems [20], zero entries are meaningful, that is, they are comparable with very small matrix entries. This situation enables the use of the popular SVD-based methods. In the rating decision problems, however, zero has a significantly different meaning: it reflects an *event* of a missing data. Thus, once we define *event recovery* as a main goal, a better representation of data can be a set of events (that one can achieve with the binary structures Ω_k) rather than unified in one matrix numerical values.

3.2 Randomized Filtering of the Training Set

To demonstrate the power of the new method over the widely used linear methods, we introduced additional uncertainty to the system by filtering out about 85% of known data from the originally defined Ω_k. A nonzero value related to the pair (m_p, u_q) was removed from the corresponding Ω_k with probability 0.85 if the total number of values in the respective row $\Omega_k(p,:)$ was greater than some sufficiently small threshold (in the presented experiments, 5).

As a result of this filtering, one can observe the clear correlation between the prediction quality growth and the increase of interpolation order. Moreover, the running time of these experiments was significantly faster and easily fit the memory of personal computers.

In our work, we do not use the available sophisticated methods of filtering that minimize the loss of information. Our goal is not to compress the data for optimal storage, but rather to account for as many prediction uncertainty factors as possible, and so to demonstrate the benefits of high-order interpolation.

The experimentation with very low memory expenses also had another purpose. In many highly parallel architectures the amount of memory one can allow for each processor is much smaller than the size of the original Ω_k and, thus, of SVD components of $\Omega_k \Omega_k^T$ (especially when the distribution of non-zeros is close to power law distribution). It was important, therefore, to check the ability of our method to work on such architectures.

3.3 Latent Factors Identification

We have constructed Ω_k with the expectation that information about each *known* decision is sufficient to also recover the *unknown* decision. Each known decision, however, cannot be treated as a separate input of θ since there are too many of them. Even after filtering the training set, the number of entries in each row of Ω_k is clearly too large to construct even a linear approximation based on the available limited number of relevant ratings.

We need to extract only a few principal factors influencing the likelihood of rating r_k. We will use the principal factors identified by the standard singular value decomposition [19] of Ω_k:

$$\Omega_k = \Phi \cdot \Sigma \cdot \Upsilon^T = \sum_i \phi_i \sigma_i \, , \upsilon_i^T \tag{5}$$

where σ_i are the singular values listed in ascending order. It is well known that the optimal lower-rank approximation of Ω_k is a truncated version of the decomposition (for details see [19,21]):

$$\hat{\Omega}_k = \hat{\Phi} \cdot \hat{\Sigma} \cdot \hat{\Upsilon}^T = \sum_{i=1}^{\eta} \phi_i \sigma_i \upsilon_i^T \, , \tag{6}$$

where η is a desired number of principal factors (for the details on choosing η see Section 5). The projection of query q (i.e., q is a column of Ω_k that describes u_i with corresponding missing entry) onto a lower-dimensional subspace with the basis determined by SVD is defined as

$$\hat{q} = \hat{\Phi} \hat{\Sigma} q \in \mathbb{R}^{\eta} \, . \tag{7}$$

To define the subspace, we need only the left singular vectors ϕ_i. We find them by solving an eigenvalue problem

$$C\phi_i = \sqrt{\sigma_i}\phi_i \ , \tag{8}$$

where $C = \Omega_k \Omega_k^T$ is the covariance matrix of Ω_k. Let $(\phi_1, \phi_2, ..., \phi_\eta) = \hat{\Phi}$ be the first η columns of Φ, that is, the eigenvectors corresponding to η largest eigenvalues. The factors $S = (s_1, s_2, ..., s_\eta)$ determining the likelihood of rating r_k to unit of content m_j are defined as vectors

$$S = \hat{\Phi} \cdot \text{diag}(\sqrt{\sigma}_1, ..., \sqrt{\sigma}_\eta) \cdot \Omega_k(:, j), \tag{9}$$

and the desired polynomial (instead of the linear) decision-making function θ is constructed as an expansion

$$\theta = \sum_j x_j \psi_j(S) = \sum_j x_j \psi_j(s_1, s_2, ..., s_\eta) \tag{10}$$

with the polynomial basis $\Psi = \{\psi_j(S)\}$ and coefficients $x_j \in \mathbb{R}$ (see the next section).

We note that more complex approaches to information reduction than straightforward SVD-decompositions exist. For instance, [22,23,24] offer elegant methods for matrix factorization with missing entries.

3.4 Polynomial Approximation on Latent Factors

A polynomial expansion is a generic way to describe the response of the model where all inputs are known scalar values, and the output is also a scalar. The idea is to initially assume equal importance of all inputs, list all possible combinations of all input variables to all positive integer powers, and build a multi-variable polynomial basis using all such combinations, up to some maximal total order in each polynomial. The output is then represented as an expansion in this polynomial basis, with the expansion coefficients obtained by regression on the available training set of inputs and outputs. For textbook information on such polynomial decomposition (originally known as *polynomial chaos*) see [25]). We note that multi-variable functions other than polynomials may also be used to build an approximation.

We define a multivariable polynomial basis $\Psi = \{\psi_j\}$ by

$$\psi_j(S) = \psi_j(s_1, s_2, ...s_\eta) = \prod_{l=1}^{\eta} p^{(\tau_l)}(s_l) \tag{11}$$

with single variable polynomials $p^{(\tau_l)}$ of order τ_l. For example, if there are only two latent factors, s_1 and s_2, and the maximal interpolation order is 2, then the basis will consist of the polynomials

$$\begin{array}{l} \psi_1(S) = p^{(0)}(s_1)p^{(0)}(s_2),\ \psi_2(S) = p^{(1)}(s_1)p^{(0)}(s_2), \\ \psi_3(S) = p^{(0)}(s_1)p^{(1)}(s_2),\ \psi_4(S) = p^{(2)}(s_1)p^{(0)}(s_2), \\ \psi_5(S) = p^{(1)}(s_1)p^{(1)}(s_2),\ \psi_6(S) = p^{(0)}(s_1)p^{(2)}(s_2). \end{array} \tag{12}$$

A straightforward choice is a trivial basis $p^{(i)}(\alpha) = \alpha^i$, leading to the following construction.

Total order:	Polynomials:
0	1
1	$s_1, s_2, s_3, ...$
2	$s_1^2, s_1 s_2, s_1 s_3, ..., s_2^2, s_2 s_3, ...$
3	$s_1^3, s_1^2 s_2, s_1^2 s_3, ..., s_1 s_2 s_3, s_1 s_2 s_4, ...$

The expansion coefficients x_i are obtained by solving the linear regression equations

$$\sum_l x_l \psi_l(S) = \mathcal{R}(u_i, m_j, r_k) \; . \tag{13}$$

This system of linear equations has as many right-side entries as there are known decisions of the current user. At the same time, there should be at least one equation per polynomial in the basis. The number of required known decisions (for efficient utilization of basis Ψ) grows combinatorially with an increase in the number of factors and the maximal polynomial order. Since only a small fraction of content was rated by the current user, the decision-making function has to be constructed using a small number of factors and a polynomial basis of low order. We can allow a slight increase in the number of factors by using an incomplete basis, where only some variables are included in the polynomials of higher degree.

3.5 Application-Oriented Postprocessing

In general, the best way to recover $\mathcal{R}(u_i, m_j, r_k)$ depends on the particular application for which θ is evaluated. We experimented with different postprocessing strategies to finalize the discrete prediction. A straightforward postprocessing approach (with corresponding results in Table 1) consists of a weighted averaging in which a final rating is calculated by

$$r_{predicted} = \frac{\sum_{k=1}^{K} r_k \theta(u_i, m_j, r_k)}{\sum_{k=1}^{K} \theta(u_i, m_j, r_k)} \; . \tag{14}$$

Algorithm 2. Detailed Algorithmic Scheme Used for Obtaining Presented Numerical Results

For each r_k define Ω_k as in (4)
Introduce the additional uncertainty as explained in Section 3.2
For each $r_k \in R$ design corresponding $\theta(u_i, m_j, r_k)$ by
    Define the covariance matrix $C = \Omega_k \Omega_k^T$
    Obtain eigen-system $\hat{\Phi} = (\phi_1, \phi_2, ..., \phi_\eta)$
    Obtain principal factors $S = \hat{\Phi} \cdot \text{diag}(\sqrt{\sigma}_1, ..., \sqrt{\sigma}_\eta) \cdot \Omega_k(:, j)$
    Define a polynomial basis $\Psi = \{\psi_j\}$ as in (11)
    Obtain coefficients x_l for $\theta(u_i, m_j, r_k) = \sum_l x_l \psi_l(S)$ by
        solving the system (13)
Return $r_{predicted} = \sum_{k=1}^{K} r_k \theta(u_i, m_j, r_k) / \sum_{k=1}^{K} \theta(u_i, m_j, r_k)$

More advanced ways to determine a final prediction are discussed in Section 5. Algorithm 2 is a detailed scheme used for obtaining the presented numerical results.

4 Experimental Results

The numerical experiments were performed on randomly chosen data points (u_i, m_j, r_k) from a Netflix contest database [8]. During each experiment $2{\cdot}10^5$ points were extracted from a training set, and their ratings were predicted. We used standard LAPACK implementations for eigenvalue decomposition (over 60% of computational cost) and other linear algebra operations. We did not alter our method to optimize the running time.

The first portion of experiments was designed to check the quality of the data compression and prediction methods, when the query point was not eliminated. The demand of the almost sure closeness to the unknown $\mathcal{F}(q)$ in 1 was completely approved. The partial (visible) results are presented in Figure 1. In this particular subexperiment we investigated 1,000 queries (represented on x-axis). For each query, the corresponding Ω_k for true r_k was taken as an input for θ. All the recovered values (events of assigned rating) concentrated around 1, demonstrating that the data compression technique does not destroy this information.

The situation with the real-life queries (i.e., queries with missing ratings), is different. Consider the example with 20 queries presented in Figure 2. Every query has 5 checkpoints (for each rating from 1 to 5) presented as diamond, square, triangle, circle, and star, respectively, in other words, all Ω_k were used for the corresponding decision-making functions. Let us concentrate on the 10th query. The most likely ratings that the corresponding θ has recovered are 4 (the circle at the 10th column) and 5 (the star at the 10th column). According to θ on this query at $r = 3$, it is less likely that the real rating was 3 (the triangle at the 10th column). The ratings 1 (the square at the 10th column) and 2 (diamond at the 10th column) are almost unlikely, according to their corresponding θ functions. From this example we conclude that the real rating is very likely to be 4 or 5, since the unlikeliness of 1, 2 and 3 is also confirmed.

Usually, the distribution of predictions in real queries is not as easily interpreted (see other queries in Figure 2). Thus, we need a postprocessing procedure to convert these recovered values into a real prediction. Many algorithms can be used to tackle this

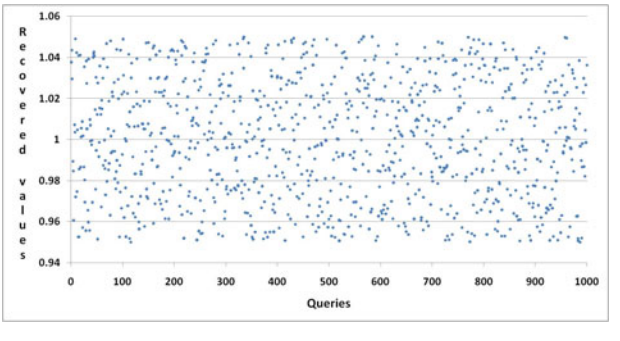

Fig. 1. Example of the *not removed* data recovery

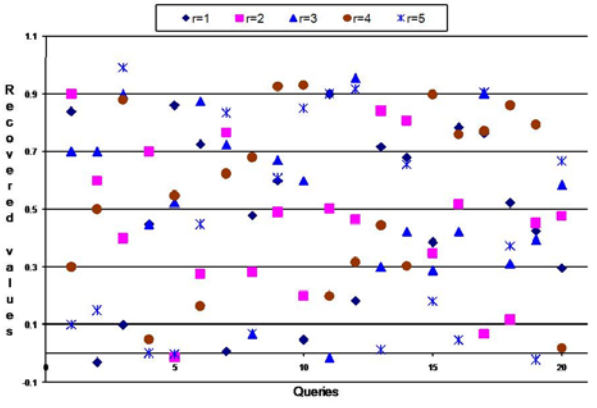

Fig. 2. Example of "real-life" recovered values

problem [20]. To concentrate on the essential and clear difference between polynomial and linear methods, we used the most simplified final rating calculation (14). A rooted mean-square error (RMSE) was suggested by Netflix as a measure of prediction success and defined as

$$RMSE = \sqrt{\frac{1}{n}\sum (r_{true} - r_{predicted})^2}. \tag{15}$$

To get an upper bound for the prediction, we first measured the RMSE obtained using simple averaging of the ratings given by all users. This approach gave an RMSE upper bound of 1.05.

A *linear* method (the interpolating basis consisted of only linear functions, an approach approximately equivalent to generic SVD-based techniques used by many of the Netflix contest participants) improved the averaging method by approximately 10%. Since linear method may depend on the number of latent factors, we experimented with $5 \leq \eta \leq 30$. The observed variability of RMSE was approximately 0.5%.

The next series of experiments was performed with higher interpolation orders, namely, 2 and 3. We observed an improvement of the final RMSE by 0.05 and 0.1, respectively. For the Netflix problem, using even higher orders of interpolation severely restricted the number of factors. Also, higher orders often caused numerical instability for several data points (thus influencing the whole postprocessing procedure). In principle, such numerical instability can be eliminated by various strategies (e.g., preconditioned collocation [26]) that are beyond the scope of current work.

The average results over 20 series of the experiments (each with $2 \cdot 10^5$ queries) on highly filtered data (see Section 3.2) are presented in Table 1. Clearly, our current numerical results are worse than those obtained by the best contest participants. As a basic information matrix (for SVD and polynomial/linear latent factor analysis) we used only a user-centered matrix that is significantly less informative than a full matrix, a combination of user- and content-centered data, and different blending approaches that use full contest information. On the other hand, such a small base information matrix significantly reduced the running time of our experiments. We note that our main goal

Table 1. Final RMSE results

Algorithm	Order	Average RMSE
Average rating	-	1.05
Linear approximation	1	0.98
Polynomial interpolation	2	0.95
Polynomial interpolation	3	0.90

is to present the potential of working with high-order polynomial interpolation that can successfully extend a popular linear approach.

5 Discussion

In this section, we introduce a number of clarifications on the motivation, development, and possible extensions of the proposed method. The remarks are presented here in the order in which the corresponding topics appeared in the previous sections.

Singular Value Decomposition. Our approach uses principal factor identification via singular value decomposition. We now clarify the relationship to the more general SVD-based prediction methods.

As in many other approaches, we use SVD to approximte the training set by projecting it onto a space of lower dimension. However, we do not use this projection as the final prediction tool, that, for example, fills in the missing ratings optimally close to this lower-dimensional subspace. Instead, we use data compression *to define the inputs of the decision-making function* θ that are optimally close to the lower-dimensional subspace.

We assume that an answer to a query is based on correlations in the training set. However, we additionally refine the approach and state that an answer to a query is an output of a deterministic nonlinear function applied to observed correlations. Our prediction, then, is not nessesarily optimally close to the lower-dimensional subspace. Instead, it is optimally close to the deviation from the lower-dimensional subspace observed for the current user.

In comparison with the general case, our method is also computationally cheaper, since we use only a small portion of the training set, filtered to increase sparsity.

We conclude that our approach is a special case of a more general group of the methods that use singular value decomposition on the training set.

Binary Matrix. The choices made in constructing the binary matrix Ω_k are an important part of our method. While filling out the entries corresponding to known decisions is straightforward, the rules for representing a *missing decision* are subject to discussion.

In principle, we suggest two options that are consistent with the probabilistic interpretation of the entries in the binary matrix. They are to either fill the uncertain entries with zeros, to represent a simple fact that the rating r_k was not given, or with white noise, or a set of uniformly distributed random numbers in the interval $(0, 1)$, to represent a fact that the likelihood of the rating r_k is uncertain. We chose the former option:

for computational efficiency, and to avoid real-valued entries into the binary matrix that was declared to store events rather than numbers.

The choice of zeros to represent uncertain ratings will have an unfortunate consequence for the unit of content m_j currently being predicted. From the examination of $\Omega_1, \Omega_2, ..., \Omega_K$ it appears that the decision being predicted is already included in the training set; and the probability of every possible outcome of the decision is 0. The decision-making functions trained on such matrices will in most cases return a low probability for every rating. We compensate for the situation by augmenting a binary matrix Ω_k with the entry $\Omega_k(j, i) := 1$.

Choice of Polynomial Basis. Use of multivariate polynomials to approximate an unknown response is based on the previous work using stochastic finite element methods, or SFEM. The approach, described in [27,28], uses a multivariate basis (also known as polynomial chaos) of polynomials that are orthogonal in some probabilistic measure. The goal of SFEM is to produce convenient expressions for the statistical distribution of the output, given some statistical information on the inputs.

Since the task in our applied problem is to reproduce the value of the output rather than its statistical properties, we are not a priori restricted to a particular polynomial basis. We have tried using several standard solutions from polynomial interpolation theory, including Chebyshev and Hermite polynomials. This led to approximately equivalent results, with a slight improvement corresponding to a trivial basis.

The number of the polynomials allowed in the basis is limited by the number of known decisions of the user. If we use a full basis of fixed maximal order, the number of the required decisions grows quickly. For example, for a basis of order 2 on 30 variables we need at least 496 collocation points; for a basis of order 3 on 30 variables we need 5,456 points.

In the context of our applied problem, where each user rated maybe 500 units of content, the use of large numbers of factors is unrealistic. In practice, we are limited to polynomial basis of orders 2 and 3, on as many variables as we can fit to the available number of known decisions.

A tradeoff is possible: in our experiments we compared the performance of the method on a basis of higher order with a few variables versus a lower order with more variables. The basis of order 3 resulted in a consistent improvement, indicating that nonlinearity of the response is a more important feature of the problem than incorporation of more degrees of freedom. With use of the basis of order 4, however, the quality has sharply decreased, because of the numerical instability of the collocation matrix and an extreme limitation on the number of variables.

Parameters. We note that the suggested approach is free of the hidden parameters that principally influence the performance of the method. Once the polynomial basis (of a particular order) has been selected, the maximal number of factors that can be taken into account is set deterministically (from a lookup table); using fewer factors amounts to switching to a smaller basis.

Improving Prediction Quality. In our experiments, we have used a number of postprocessing strategies to recover the decisions based on the values of the decision-making

function. Many of the options did not immediately improve the performance but present interesting avenues for further study of the method.

In particular, we used different modifications of the general weighted strategy

$$r_{predicted} = \frac{\sum_{k=1}^{K} r_k w_k}{\sum_{k=1}^{K} w_k}, \tag{16}$$

with weights w_k (in 14 $w_k = \theta(u_i, m_j, r_k)$). In some tests, better results were achieved by a version corrected by $F_k = |\{u \in U : \mathcal{R}(u, m_j, r_k) = 1\}|$, the size of the group of the users that agreed on the rating r_k for the current movie:

$$w_k = \theta(u_i, m_j, r_k) F_k. \tag{17}$$

The prediction can be further reinforced by introducing various correction variables for the values of θ and F_k. One approach is a simple fixed-point iteration technique, introducing a correction term Δ at each step:

$$w_k = \theta(u_i, m_j, r_k)(1 + \Delta). \tag{18}$$

The correction term can be derived by applying the procedure to predict the already known decisions. We have also attempted a polynomial structure, including several correction terms:

$$w_k = \theta(u_i, m_j, r_k)(1 + \Delta_0 + \sum_{l=1}^{L} \Delta_l \theta(u_i, m_j, r_k)^l) \tag{19}$$

Again, at each step we have obtained the values of Δ_l by interpolation, using the predictions of known decisions as training points.

In addition to occasional improvement in performance, the suggested reinforcements open several directions for making the general prediction framework more flexible. For example, the approach can be easily combined with steepest descent learning procedures. Since the polynomial interpolation predicts the probability of an outcome, rather than a numerical value, the prediction method can be combined with many of the existing artificial intelligence approaches generally based on fuzzy logic.

6 Conclusions

In this paper we generalize a widely used method for working with the latent factors of an information model. The generalization consists of a high-order polynomial interpolation scheme rather than linear combination. The presented algorithmic approach is highly adaptive and can be reinforced by iterative parameter learning methods.

In addition, for a particular class of rating-based applications of a recommender systems, we introduced an event matrix model as a baseline for a latent factor methods, which can describe better a fact of missing data and successively interact with the high-order polynomial scheme.

The experiments on data reinforced by introduction of additional aggressive uncertainty exhibited significant improvement in comparison to the linear method and an improvement produced by an increase in interpolation order from 2 to 3.

Although this method does not represent a final unique recipe and must be tuned according to the application needs, it can determine a main algorithmic strategy. Overall, the method appears to be competitive in its class, requires a moderate implementation and computational cost, and can be combined with sophisticated post-processing techniques. We recommend considering the high-order polynomial interpolation scheme for data recovery and prediction algorithms that are based on latent factors extraction.

References

1. Brand, M.: Fast online svd revisions for lightweight recommender systems. In: Barbará, D., Kamath, C. (eds.) SDM. SIAM, Philadelphia (2003)
2. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Incremental singular value decomposition algorithms for highly scalable recommender systems. In: Fifth International Conference on Computer and Information Science, pp. 27–28 (2002)
3. Berkhin, P.: A survey of clustering data mining techniques. In: Grouping Multidimensional Data, pp. 25–71 (2006)
4. Kumar, V.: Data Mining for Scientific and Engineering Applications. Kluwer Academic Publishers, Norwell (2001)
5. Azar, Y., Fiat, A., Karlin, A., McSherry, F., Saia, J.: Spectral analysis of data. In: STOC 2001: Proceedings of the thirty-third annual ACM symposium on Theory of computing, pp. 619–626. ACM, New York (2001)
6. Jacsó, P.: Google Scholar: the Pros and the Cons. Online Information Review 29(2), 208–214 (2005)
7. Adamic, L., Adar, E.: How to search a social network. Social Networks 27(3), 187–203 (2005)
8. Netflix prize problem, `http://www.netflixprize.com/`
9. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. on Knowl. and Data Eng. 17(6), 734–749 (2005)
10. Armstrong, J.S.: Principles of forecasting: A handbook for researchers and practitioners. Kluwer Academic Publishers, Dordrecht (2002)
11. Burke, R.: Knowledge-based recommender systems. In: Kent, A. (ed.) Encyclopedia of Library and Information Systems, Marcel Dekker, New York (2000)
12. Talwalkar, A., Kumar, S., Rowley, H.: Large-scale manifold learning. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
13. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290(5500), 2319–2323 (2000)
14. Roderick, O., Anitescu, M., Fischer, M., Yang, W.S.: Polynomial regression approaches using derivative information for uncertainty quantification. In: Nuclear Science and Engineering (2009) (to appear)
15. Roderick, O., Anitescu, M., Fischer, M., Yang, W.: Stochastic finite-element approach in nuclear reactor uncertainty quantification. American Nuclear Society Transactions 100, 317–318 (2009)
16. Herlocker, J., Konstan, J., Terveen, L., Riedl, J.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems 22, 5–53 (2004)

17. Resnick, P., Varian, H.: Recommender systems. ACM Transactions on Information Systems 40, 56–58 (1997)
18. Gorsuch, R.L.: Factor Analysis. Erlbaum, Hillsdale (1983)
19. Dym, H.: Linear Algebra in Action. American Mathematical Society, Providence (2007)
20. Christopher, D., Manning, P.R., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
21. Berry, M.W., Dumais, S.T., O'Brien, G.W.: Using Linear Algebra for Intelligent Information Retrieval. SIAM Review 37(4), 573–595 (1995)
22. Zhang, S., Wang, W., Ford, J., Makedon, F.: Learning from incomplete ratings using non-negative matrix factorization. In: Proceedings of the SIAM Conference on Data Mining (2006)
23. Schein, A.I., Saul, L.K., Ungar, L.H.: A generalized linear model for principal component analysis of binary data. In: Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics (2003)
24. Brand, M.: Incremental singular value decomposition of uncertain data with missing values. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 707–720. Springer, Heidelberg (2002)
25. Ghanem, R., Spanos, P.: The Stochastic Finite Element Method: A Spectral Approach. Springer, New York (1991)
26. Bellomo, N., Lods, B., Reveli, R., Ridolfi, L.: Generalized Collocation Methods. Birkhauser, Basel (2008)
27. Ghanem, R., Spanos, P.: Polynomial chaos in stochastic finite elements. Journal of Applied Mechanics 57, 197 (1990)
28. Spanos, P., Ghanem, R.: Stochastic finite element expansion for random media. Journal of Engineering Mechanics 115(5), 1035–1053 (1989)